C++学习笔记总结1，2，3

2022年3月19日 20:59

- C语言知识：变量类型，数组，函数，指针，递归
- C++
  - 容器/类容器：
    - string
      - 定义
        - string是一个用来存放字符串的类容器，可以简单的理解为一个新的变量类型，支持变量类型的所有操作
      - string类对象支持迭代器和下标操作，迭代器解引用的对象就是单个字符
      - 初始化
        - 默认初始化：是一个空串
        - 初始化的时候可以使用圆括号也可以使用花括号，分别代表不同的类型
          - 圆括号：直接初始化，拷贝初始化
            string s（”hello“）/string s=（”hello“）
          - 花括号：可以将两个字符连起来，也可以进行双引号的拷贝初始化
            string s={‘s','s'}/string s{“hello”}
      - 输入
        - 有两个函数可以去输入
          - cin
            - 用法：cin>>s
            - cin可以读入串字符，遇到空格时候停止
          - getline
            - 用法：getline（cin，s）
            - 可以读取一整行（遇到换行符停止，包括空格）
      - 操作（除了输入输出）
        - 迭代操作
          - 利用迭代器进行操作，迭代器有自己的类型，但是一般用auto来代替
            auto c=s.begin() / auto d=s.end() 其中c，d为迭代器
        - 下标操作
          - 利用下标直接对字母进行随机访问，也是可以的
            char c=s[3];
        - 范围语句for循环操作
          - 搭配auto，我们可以进行范围内循环
            for（auto c：s）。。。。。
          - 如果要更改字符串中字符的值，auto需要为引用类型

- 加操作
  - 可以让两个字符串相加
    s=s1+s2
  - 可以让两个字符串和一个字面值相加，但是字面值的两边是必须要有字符串的，否则就会报错
    s=s1+“a”+s2
- 注意
  - 在使用循环的时候，我们通常使用迭代器进行操作，而不使用下标，原因是下标在大多数标准库不被支持，而迭代器基本支持每一个标准库

- vector
  - 定义
    - vector是一个可以无限储存东西的容器
  - 初始化
    - vector默认初始化：什么也不放，也最建议
    - 括号初始化
      - 执行直接初始化功能
        vector v2（v1）
      - 对于括号初始化，我们更加倾向于数量，对于一个数字，我们倾向于建立空格
        vector v（10）v是一个已经有10个空格的vector
    - 列表初始化
      - 我们更加倾向于向里面放元素
        vector v{20} v里面有一个元素。这个元素的数字是20；
  - 输入
    - v.push_back(i->对应的数字)
  - 操作v
    - 迭代操作
      - 利用迭代器进行操作，迭代器有自己的类型，但是一般用auto来代替v
        auto c=v.begin() / auto d=v.end() 其中c，d为迭代器
    - 下标操作
      - 利用下标直接对字母进行随机访问，也是可以的v
        int（这里指对应的类型） a=v[3];
    - 范围语句for循环操作
      - 搭配auto，我们可以进行范围内循环
        for（auto c：v）。。。。。
      - 如果要更改字符串中字符的值，auto需要为引用类型

- 标准库
  - cctype：处理字符使用
  - cstdio：c语言部分函数的标准库
  - cassert：assert调试函数的处理哭
  - initializer_list：抛出异常的标准库
    try{}catch（xxx）{}
- decltype/auto的介绍
  - decltype
    - decltype是一种特殊的类型，因为它可以“猜”出来一个变量的类型，如果你需要使用变量的类型，但是不清楚变量的类型是什么时候，就使用decltype
      decltype（p） s=2
    - p可以是任何类型，指针，函数，等等其他的
    - 对于返回函数指针/数组等类型我们叫做尾置返回类型
  - auto
    - auto也是一种特殊的类型，它可以将一个对应的类型赋予
      auto a=2；//赋予a：int类型
      auto func（）->int(*)(int,int)//赋予func右边的类型（函数指针类型）
    - 对于返回函数指针/数组等类型我们叫做尾置返回类型
- 特殊的函数操作
  - 函数重载
    - 定义
      - 对于C++里面两个函数的名字相同，但是对于参数类型不同，参数数量不等的函数是可以存在的，我们把它叫做函数的重载
      - 注意：对于两个函数名字相同：只看参数：参数数量不同/类型不同二者之一即可重载，否则不可以
      - 重载的函数可以减少程序员记名字的负担
  - 函数指针
    - 定义：指向函数的指针
    - 创建：把函数的名字改成（*xx）xx为指针的名字，且xx就是那个指针了
      int（*P）（int a，int b）
      - 注意这里：函数的指针和其他指针一样，都有类型和指向
        - 类型
          - 指针的类型包括两部分，参数和返回值，也就是说如果两个指针都相同必须这两个因素也相同
            比如上述指针的类型就是int （*）（int ，int ）类型，函数指针
        - 指向
          - 指针必须具体指向那个函数，或者不指向函数，引用符可选
            int（*P）（int a，int b）=a//a是函数且类型一样

int（*P）（int a，int b）=nullptr//空指针

- 传入函数指针
  - 可以直接把函数的名字当作指针传入
    sss(2,3,函数名)
- 调用函数
  - 指针可以充当函数，在指向明了之后可以直接调用函数
    P（2，3）
- 声明返回函数指针类型的方法
  - 与数组类似
    - 注意decltype猜测的时候与auto略有不同，decltype猜测为数组，auto为头指针
  - 1.typedef/using方法
    typedef int FUN（int a，int b）；（FUN新名字替代原名字位置，a，b可以省略）
    FUN* C(int i)

    using FUN= int fun（int a，int b）；
    FUN* C（int i）

  - 2.decltype/auto
    - int fun（int a）
    - decltype
      decltype（fun）*p
    - auto
      auto f（int）->int（*）（int，int）